TELKOMNIKA, Vol.17, No.2, April 2019, pp.728~737
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i2.11752

Data stream mining techniques: a review

Eiman Alothali, Hany Alashwal*, Saad Harous
College of Information Technology, United Arab Emirates University, Al-Ain United Arab Emirates
*Corresponding author, e-mail: halashwal@uaeu.ac.ae

Abstract

A plethora of infinite data is generated from the Internet and other information sources. Analyzing this massive data in real-time and extracting valuable knowledge using different mining applications platforms have been an area for research and industry as well. However, data stream mining has different challenges making it different from traditional data mining. Recently, many studies have addressed the concerns on massive data mining problems and proposed several techniques that produce impressive results. In this paper, we review real time clustering and classification mining techniques for data stream. We analyze the characteristics of data stream mining and discuss the challenges and research issues of data steam mining. Finally, we present some of the platforms for data stream mining.

Keywords: *classification, clustering, data stream mining, real-time data mining*

1. Introduction

The revolution of the information technology has created a massive amount of data stream and brought frontward the challenge of understanding the hidden patterns that it has to the attention of academia and industrial communities. The requirement of mining stream data in real time becomes essential to obtain such valuable knowledge. Data stream mining refers to mining the data on real-time by storing, processing, and extracting feasible knowledge that can help in decision making and understanding some phenomenal events. For example, in social media networks people are becoming more interested in using those networks for information, news and exchanging opinion on different subject matters. Such heavy usage of social networks generates massive data that has three computational issues: size, noise and dynamism [1]. This massive data in social network has opened the door to analyze and detect social communities, sentiment analysis, and ranking influencing users, predicting future trends and understanding social networks graphs.

Many studies focus on detecting and studying different events in social networks by using stream data mining techniques. The authors in [2], present a distributed system that performs real time sentiment analysis. They based their solution on Vertical Hoeffding Tree. The work in [3], presents an online framework for Twitter's stream that detects, clusters and tracks events using electric fields analogy. In [4], the authors use Twitter stream to detect real-time traffic events by using classification analysis. Generally, different analysis techniques that can help in studying streams data by applying different clustering and classification techniques are proposed for dynamic and static mining such as MOA and SAMOA [5, 6]. Therefore, this paper reviews data stream mining techniques that include clustering and classifications algorithms in order to help later in studying social network in particular. This paper is organized as follows: section 2 presents data stream mining characteristics. Section 3 reviews stream mining techniques with two subsections discussing clustering and classification techniques. Section 4 highlights the performance measures in data stream mining. Section 5 presents different platforms and tools for data stream mining. Section 6 addresses the challenges of these techniques and future direction.

2. Data Stream Mining Characteristics

Data stream mining has many unique characteristics that make a contrast when compared to traditional data mining as shown in Table 1. Many researches have addressed these characteristics of data stream in details such as [7, 8]. In general, we summarize these characteristics in three main points.

Received June 3, 2018; Revised October 22, 2018; Accepted November 19, 2018

a. Size: The large volume of data stream makes it impossible to store such data for further mining. This infinite data required obtaining valuable information from data mining with one pass scan. The massive data stream has different data forms such as texts, images and other. Therefore, only a synopsis of the data stream mining is saved.
b. High speed: The high speed of generating this massive data makes a high requirement for efficiency of data mining. The dynamic arrival of massive data stream changes over time. Therefore, it may be impacted by factors such as measurement or calculation model errors.
c. Multidimensional: Massive stream data are produced from heterogeneously distributed sources with different data types. Therefore, it requires sophisticated algorithms to mine. There are great difficulties due to the large volume of stream data that allows process and computation in one pass.

Table 1. Comparison between Traditional Data Mining and Stream Data Mining

Feature	Traditional Data Mining	Data Stream Mining
Processing	Offline every record	Real-time samples
Storage	Feasible	Not feasible
Volume	Finite	Infinite
Data generation	Static	Dynamic
Time	More time to access data	Only one pass
Data Type	Homogeneous	Heterogeneous
Result	Accurate	Approximation

3. Data Stream Mining

In data streams, the infinite flow and the speed sequence of instances make the process challenging. Data stream mining techniques are required to effectively analyze, integrate, obtain, and transform data to extract valuable patterns in real time in only a single scan and maintain continuity of process. In this section we review some of the real-time clustering and classification data stream mining techniques.

3.1. Stream Clustering Techniques

Data stream clustering has been the subject of attention in research due to its effectiveness. Adapting an arbitrary clustering algorithm to data streams is difficult as there is only one scan pass. Stream clustering can be performed for the whole stream or by using sliding window where only last recent items of the stream is considered. Several clustering algorithms and methods have been proposed and discussed by researchers [9, 10]. In this section we review some clustering algorithms for real-time clustering data streams as shown in Table 2. These algorithms are categorized based on the clustering methods they follow.

3.1.1. Partitioning Method

A common clustering technique is partitioning clustering methods using k-mean and k-median. The goal is to classify objects into clusters based on similarity. The number of clusters to be generated has to be predefined by user. CluStream [11] and HPStream [12] are partitioning algorithms using two components of clustering method; online micro-clustering and offline macro-clustering. The online component stores the summary statistics of stream using very efficient process for storage. In CluStream a pyramidal time pattern is used to store micro-clusters at snapshots based on their arrival timing. In HPStream, a projection based clustering is used to perform efficiently when applied to a high dimensional data to solve this issue in CluStream. HPStream algorithm is capable of handling high-dimensional cases. However, obtaining an appropriate average projection dimension is difficult.

DCSTREAM [13] is a clustering algorithm based on Divide-AND-Conquer using k-means to generate micro-clusters. Based on the samples length in vector model, it captures the general structure of the clusters without requirement to store the complete clusters in memory. The online component consists of three modules: subsets generator, micro-cluster generator and split and merge module. Handling concept drift during the process is maintained by split and merge module.

SODA [14] is a recent partitioning algorithm for static data but it has an extension for clustering data stream. It can continuously process data streams based on the offline

processing of an initial dataset. The partitioning of data streams starts with a brief offline data set and follows the changing data pattern in an active manner once initialized with a seed dataset. SODA is able to perform clustering autonomously with very high computation efficiency and produces high quality clustering results.

Table 2. Summary of Surveyed Clustering Data Stream Mining Techniques

Algorithm	Year	Method	Key Feature
CluStream [11]	2003	Pyramidal time frame	Online micro-clustering and offline macro-clustering using pyramidal time pattern in online phase and k-mean in offline phase
HPStream [12]	2004	Projection based	Projection based clustering for high-dimensional streaming data
DCSTREAM [13]	2016	Divide and conquer	Dividing samples based on their length within the vector model to capture the general structure of the clusters without requiring the complete cluster data to be stored in memory
SODA [14]	2018	Partition-based	Using nonparametric empirical data analytics to demonstrate high clustering and efficiency
DenStream [15]	2006	Density-based	Using density connectivity concept, the set of micro clusters are used to create global clustering
D-Stream [16]	2009	Density-based	It creates a set of disjointed grids by dividing the data space.
CEDAS [17]	2017	Graph-based	Two stages technique for clustering evolving data stream in a way that keeps minimal and simple calculation using hyper-spherical micro-clusters.
FStream [18]	2018	Density-peak	Based on the fast density peak search method that makes it suitable for large streams since it doesn't require any iterations in its implementation
ODAC [19]	2008	Hierarchical Divisive-Agglomerative	Incremental clustering stream time series that constructs a hierarchical tree-shaped structure of clusters using top-down approach
E-Stream [20]	2007	Hierarchical agglomerative	Classify five types of evolution: appearance, disappearance, self evolution, merge and split for improved detection of changes in data stream clustering
HUE-Stream [21]	2011	Evolution-based	An extension of E-Stream to support uncertainty in heterogeneous data using distance function
CluDistream [22]	2007	EM-based	An Expectation Maximization algorithm with Gaussian mixture model to learn the underlying data distribution
SWEM [23]	2009	Time-based sliding window	Using time-based sliding window approach with the expectation maximization technique
SNCStream [24]	2015	Social-network theory	One step online clustering that is capable of finding non-hyper-spherical clusters
AFTER-STREAM [25]	2017	Model-based	Using Chebyshev statistical test to detect outliers and merge compatible clusters

3.1.2. Density-based Method

Density-based algorithms are effective in detecting arbitrary shapes clusters and handling noise in data. These algorithms do not require a predefined number of clusters and do not work well in multidimensional data. DenStream [15], D-Stream [16], CEDAS [17] are a density-based algorithms for clustering evolving data streams. Both DenStream and D-Stream use a fading function to reduce the weight of each micro cluster with time and density connectivity concept to create global clustering. DenStream is able to handle arbitrary shaped clusters. It has high time complexity due to the numerous time vector computations. In other hand, D-Stream creates a set of disjointed grids by dividing the data space. Every new data is mapped into a corresponding grid based on its dimensional values. The clusters are obtained based on the density and the connectivity of grids. D-Stream is incapable of processing very high-dimensional data compared to DenStream.

CEDAS [17] works in two steps to assure accuracy, handling noise and efficiency in memory. In the first step micro-clusters are produced with a fixed small radius *r* and a simple linear aging technique to allow unused micro-clusters to die out completely. In the second stage, the algorithm searches for overlapping micro-clusters that are defined based on kernel region. Thus, any micro-clusters with minimal number of samples within the radius and have no intersection with other micro-clusters is considered as outliner micro-clusters of defined graphs of order 1 without edges.

FStream [18] is a recent clustering algorithm for large streams based on the fast density peak search method. It doesn't require any iteration in its implementation. FStream is consisted of two algorithms. The first algorithm is to find the clusters of the training set and starts afterward to assign the stream to the closest clusters. The second algorithm applies the fast density-peak-search clustering and using sliding window model to realize the stream process. It needs fewer parameters than other clustering algorithms.

3.1.3. Hierarchical Method

This method of clustering constructs a hierarchical tree shaped structure of clusters by agglomerative approach or divisive approach. ODAC [19] is an online divisive-agglomerative clustering algorithm for time series data stream. It maintains a top-down approach binary tree structure hierarchy of clusters that evolves with data. ODAC uses a correlation based dissimilarity measure for splitting each node. It handles the concept drift by detecting the change in the existing clusters and updates time and memory consumption. This algorithm computation complexity is $O(n^2)$.

E-Stream [20] and HUE-Stream [21] are evolution-based algorithms based on agglomerative method for clustering data streams. They classify five types of evolution: appearance, disappearance, self-evolution, merge and split. The clustering model starts empty then incoming data are considered isolated clusters. Once a sufficient dense region appears, the cluster is formed. Incoming data joins similar cluster based on score or classified as isolated clusters. In HUE-Stream a distance function and histogram management are used to determine uncertainty in both numerical and categorical attributes and to merge/split clusters to find nearest clusters for new data. It supports monitoring and detection of clustering structures change over time.

3.1.4. Model-based Method

Model-based clustering is a hypothesized model that provides usually a statistical way to automatically determine the number of clusters and best fits of data points while taking into account outlier points. CluDistream [22] and SWEM [23] are clustering algorithms that are based on expectation maximization method. CluDistream aims to handle incomplete data records. It fits landmark window scenarios where only insertion exits. It uses a test and cluster strategy to reduce processing cost for online clustering. In SWEM, a time-based sliding window approach is applied. The first phase is to create synopses as micro-clusters. In the second phase, synopses are used to create global clusters. It can deal with the memory limitation and handle the missing data. It performs better time complexity and quality of clusters.

SNCStream [24] is a social network-based data stream-clustering algorithm. It is based on the degree of dissimilarity of intra-cluster and inter-cluster data during the stream. It addresses the problem as a network formation and evolution problem where micro-clusters create clusters based on homophily. It is a one step online clustering that achieves high clustering quality and is unbounded by the amount of ground-truth clusters.

AFTER-STREAM [25] is an algorithm for noisy data streams. It uses statistical tests to detect outliers and merge compatible clusters. It is based on incremental online updating clustering model with the new data points and maintaining multiple cluster summaries over time. It estimates both spatial and temporal compactness of each cluster. The computational complexity of clustering is $O(NK^2)$ where K is the number of clusters at time n.

3.1.5. Discussion

Clustering algorithms for real time data stream mining propose different issues. Table 3 addresses the strengths and limitations of reviewed clustering algorithms. Clustering in real-time is a complicated task that requires handling massive data into quality of clusters while maintaining efficiency in memory and computational costs. Different factors impact clustering quality, starting from whether the processing mechanism requires a seed dataset or not, data dimensionality and performance influential factors such as parameters values and storage resources. All of these factors are based to some extend on the clustering method used. For example, in partitioning approach as CluStream, HPStream, user has to predefine the number of clusters to proceed while in density-based approach as DenStream this requirement is not needed as it generates automatically the number of clusters. Therefore, defining the correct (k) clusters for an evolving stream data for partitioning algorithms will remain an issue that impacts

quality of clusters. In addition, dealing with high dimensionality of evolving stream data is critical for different stream mining clustering algorithms. For instance, D-Stream can detect arbitrary shapes clusters with good scalability, but its processing decreases when the stream has high dimensionality.

Table 3. Strengths and limitations of Reviewed Clustering Data Stream Mining Techniques

Algorithm	Strengths	Limitations
CluStream [11]	- Detection of concept drift - Good scalability and quality of clusters	- Number of micro-clusters is predefined by user - Cannot handle the outliers or noise
HPStream [12]	- Handling high dimensional data - Scalability	- Predefined number of micro-clusters - Complexity
DCSTREAM [13]	- Memory efficiency - Handling concept drift	- Outlier detection
SODA [14]	- High quality of clusters - Computation efficiency	- Initial seed dataset
DenStream [15]	- Handling arbitrary shapes clusters - Detection of outliners	- Time complexity
D-Stream [16]	- Detecting arbitrary shapes clusters	- Time complexity when handling high dimensional data
CEDAS [17]	- Handling noise - Drift and anomaly detection	- Handling high dimensional data - Memory requirement
FStream [18]	- Fewer parameter to define	- Training set requirement
ODAC [19]	- Handling concept drift	- Computation complexity
E-Stream [20]	- Support five types of evolution	- Time complexity
HUE-Stream [21]	- Support uncertainty in heterogeneous stream data	- More parameters to define
CluDistream [22]	- Handling missing and noise data	- Based on the landmark window scenario - Parameter sensitivity
SWEM [23]	- Handling missing data - Handling memory limitation	- Parameter sensitivity
SNCStream [24]	- Quality clusters accordingly to the CMM. - Scale-free model	- Parameter sensitivity
AFTER-STREAM [25]	- Handling noisy data stream	- Computational complexity

3.2. Stream Classification Techniques

Classification techniques are the most common and well-studied technique for predictive data mining and discovering knowledge. Classification for data streams can be applied into offline and online streams [5]. Online stream classification processes and updates data as it comes one by one based on the characterization. In this section we review some classification algorithms that are used for classification mining of data streams as shown in Table 4.

3.2.1. Tree-based

VFDT [26] and CVFDT [27] are algorithms based on Hoeffding tree model for classification data stream. A usable model is available after training few examples. User should specify a threshold value to control the attribute splitting and maintaining best results for splitting less than the threshold. A statistical analysis comparison is made to drop less useful leaves and keep counts for all leave nodes in memory. To avoid higher memory consumption, pruning techniques are used. In CVFDT a sliding window is used to keep the model consistent and solve the concept drift in VFDT.

3.2.2. Rule-based

On demand classification algorithm [28] is an adoption of micro-clusters in Clustream [11]. Summary statistics are labeled on classes based on each specific micro-cluster and are updated every time new data arrives. This model uses micro-clusters classification after defining a time horizon to solve the issue of concept drift in data evolution.

SimC classifier [29] is based on instance based learning algorithms where the algorithm stores a representative subset of data itself. It works by using three features: a similarity function, the selection of the instances and the classification function. This classifier keeps a

representative small set of information in order to preserve the distribution of classes over the stream. It has better control for noise and outliers by using a removal/insertion policy.

Table 4. Summary of Surveyed Classification Stream Data Mining Techniques

Algorithm	Year	Method	Key Feature
VFDT [26] CVFDT [27]	2000/ 2001	Tree-based	Incrementing decision tree by spending very limited time on each new item. CVFDT solves the issue of concept drift
On-demand Stream Classifier [28]	2004	Rule -based	Using micro-clusters to classify them based on time horizon for better performance and concept drift
Similarity-based Data Stream Classifier (SimC) [29]	2014	Rule -based	The classifier is based on (Instance-Based Learning) and using a similarity function, the selection of the instances and the classification function
SAE2 [30]	2014	Ensemble-based	Maintains an ensemble of classifiers that are arranged as a network in which connections are created between two classifiers if they have similar predictions.
SFNClassifier [31]	2014	Ensemble-based	A dynamic scale-free network classifier with good accuracy and outperformed
CBCE [32]	2016	Ensemble-based	An online stream mining algorithm that deals with the issue of class evolution
KME [33]	2018	Ensemble-based	Handling multiple types of concept drift and the maximum use of knowledge under with few supervised labels
Anytime Nearest Neighbor Algorithm [34]	2006	Nearest neighbor	Generic framework for anytime nearest neighbor algorithms that can be interrupted anytime with the best answers in limited time.
SAL [35]	2018	Bayesian model	Active learning algorithm to cope with both concept drift and concept evolution by adapting the classification model to the changes of stream.

3.2.3. Ensemble-based

SAE2 [30] is a social dynamic ensemble classifier. It arranges the whole ensemble of classifiers as a network in which connections are created between similar predictions classifiers. SAE2 has more scalable adaptation method. It uses a maximal cliques and weighted majority voting to diminish prediction ties to limit ensemble size.

SFNClassifier [31] is ensemble-based classifier for a dynamic scale-free network. The ensemble is represented as a network to extract centrality metrics for weighted majority voting. It has comparable performance results in terms of accuracy and processing time. In other hand, CBCE [32] aims to maintain a base learner for each class and dynamically update the base with new coming data. It can quickly adjust to class evolution using under sampling method to handle the dynamic class-imbalance problem by gradual evolution of classes that happened by misclassification.

KME [33] is a recent classifier leverages supervised and unsupervised knowledge to detect concept drift and recognizes recurrent concepts. It evaluates weights of ensemble members and reuses preserved labeled instance from past blocks to enhance recognition ability. KME is a hybrid ensemble that combines chunk-based ensembles and online ensembles to handle different types of concept drift. KME can detect changes and evaluate equivalence level between concepts to cope with high-speed data streams

3.2.4. Nearest Neighbors

Anytime Nearest Neighbors (ANN) classification [34] is based on generic framework for anytime nearest neighbor algorithms with distance measure that can be interrupted anytime and gets the best answers in limited time. They use generic and special ordering heuristics techniques to sort the index of training data for ANN classifiers. The algorithm cannot be interrupted during step-up phase even that the process time is very short.

3.2.5. Statistical

SAL [35] uses a Bayesian model that allows multi-class classification without a predefine number of classes. It uses both labelled and unlabelled data for estimating the marginal and conditional distributions. This algorithm addresses the data streams challenges (such as infinite length, concept drift and concept evolution) and at the same time reduces the expected future error in online learning. It is aware of concept drift and concept evolution.

3.2.6. Discussion

Classification techniques for data streams have different issues as they deal with non-stationary data. The major issue for stream classification techniques is addressing the drift concept, as the learning model needs to be able to update itself once a change has occurred. As shown in Table 5, a number of algorithms such as [27], [28], [31], [35] have addressed detection of drift concept while others did not consider it. Other important issues to address while using classification for real-time mining are the memory and computational costs. In SAE2 [30] empirical results, it showed good overall accuracy while using less memory and processing time when compared to other classifiers. In stream classification, it is critical to understand the preprocessing requirement, in order to use the classification algorithm in an efficient manner. This can be achieved by using mechanisms to deal with noise or missing and irrelevant data during the processing time.

Table 5. Strengths and limitations of Reviewed Classification Data Stream Mining Techniques

Algorithm	Strengths	Limitations
VFDT [26]	· Pruning technique to avoid memory consumption	· Concept drift
CVFDT [27]	· Handling concept drift · Better run-time than VFDT	· Threshold value for splitting attributes
On-demand Stream Classifier[28]	· Handling concept drift	· Sensitivity of selected time-horizon parameter
Similarity-based Data Stream Classifier (SimC) [29]	· Effective for unbalanced data streams	· Handling minority classes/ outliers
SAE2 [30]	· Less memory and processing time	· Based on assumption
SFNClassifier [31]	· Adapting concept drift	· Based on assumption · Sensitivity of period parameter
CBCE [32]	· Handling dynamic class-imbalance	· Performance may decay on non-evolve classes as it emphasis on evolved classes
KME [33]	· Handling gradual and increment drifts	· Cannot handle sudden drifts in datasets with nominal attributes
Anytime Nearest Neighbor Algorithm [34]	· Good accuracy in diverse datasets	· Handling concept drift
SAL [35]	· Handling concept drift and evolution	· Class discovery performance decreases when applied to highly unbalanced data

4. Stream Data Mining Performance Measures

In data mining techniques, it is important to achieve good performance in terms of effectiveness in time and memory usage. Time and memory are usually measured by the total time spent to process the data and the amount of memory consumption. In this section, we discuss the common performance measures for real-time clustering and classification.

4.1. Clustering evaluation methods:

Evaluating clustering algorithms' results for data stream is quit challenging because it depends on the deployed evaluation measures setting. These evaluation measures can be classified into two structural measures and ground truth based measures [5]. However, there are four defined parameters that evaluate the quality of clustering [8]. These parameters are completeness, purity, sum of squared errors (SSE), and silhouette coefficient.

4.2. Classification evaluation methods:

Classification evaluation measures in online data stream mining are different from traditional mining. The cross-validation cannot be used for online mining due to the time and cost. There are two evaluation techniques for data streams classification: Holdout evaluation and prequential evaluation [5]. Holdout technique can be used for stationary data stream, but it cannot be used for cases with concept drift. The prequential evaluation is good to test unseen examples without needing a holdout set and its use for stationary data streams and cases of concept drifts.

5. Data Mining Stream Platforms

There are various data stream mining platform that facilitate creation or collection of data streams with its integrated algorithms for classification and clustering. In this section, we describe in general three of the available platforms for data stream mining as shown in Table 6.

Table 6. Platforms for Stream Data Mining

Platform	Description
MOA [5]	– A java open source software that has implemented a number of algorithms for online learning from evolving data streams with concept drift. – It provides a collection of algorithms for real-time classification, clustering and graph mining besides number of evaluation measures.
SAMOA [6]	– A java platform for stream data machine learning considered as an extension of MOA. – It can be used to develop distributed streaming machine learning algorithms and execute them in another stream-processing engine such as Spark or Storm
Apache Spark [36]	– It is good for data analytics pipeline for anomaly detection. – It includes main components such as Spark core and libraries: Sparks Machine Learning library (MLlib), GraphX for graph analysis, Spark Streaming for stream processing

6. Conclusions and Future Work

Data Stream mining problems remain an important motivation for research community to study and to propose new solutions. The volume of data increases rapidly where the existence of efficient algorithms for stream mining is important. The constraints that data stream mining has are different than traditional mining. However, remaining issues are still challenging in this area due to the characteristics of stream data. For example, clustering algorithms for one pass are scalable in general, but the issue is dealing with streams evolving. As for classification, the data size allows a constant memory per data sample because mining the entire data at one time is impossible. Concept drift is another issue because previously learned models are invalid with the evolving stream. Model updates can be costly in stream classification since it only affords constant time per data sample. In this section, we outline the main challenges for streams data mining techniques.

a. Infinite Stream: This challenge addresses the memory requirement for processing this continues huge streams of data. Storing such data is very costly and not possible. This addresses the need for more efficient techniques that will require less memory with higher accuracy results from single pass.
b. High Speed: Because of high speed of this type of data, the processing response needs to be fast and prompt in real time. This addresses the need for new algorithms that process the stream data with less computational costs and maintaining good accuracy as well.
c. Evolvement with Time: This influences the processing of new items that arrives in different timing. This active evolvement can change the shape of clusters or distribution of values over time. This calls for the need for effective techniques that identify such change.
d. Multidimensionality: Dealing with multidimensional data stream has high computation costs. It is important to reduce these costs by new effective techniques that reduce the multidimensional of stream data.

In this survey paper, we reviewed classification and clustering techniques for data stream. We discussed the strengths and limitations for clustering and classifications techniques. We addressed the main challenges and issues in data stream mining. We discussed the evaluation performance for data stream as well. Different useful tools for mining data stream were also discussed. An extension of this paper will be experimenting classification and clustering algorithms using MOA to study social network, in particularly; Twitter to build comparative analysis study.

References

[1] Adedoyin-Olowe M, Gaber MM, Stahl F. A Survey of Data Mining Techniques for Social Media Analysis. *ArXiv Prepr ArXiv13124617*. 2013
[2] Rahnama AHA. *Distributed Real-Time Sentiment Analysis for Big Data Social Streams*. In: Control, Decision and Information Technologies (CoDIT), *2*014 International Conference on. IEEE. 2014: 789–794.

[3] Terrana D, Pilato G. Detection, *Clustering and Tracking of Life Cycle Events on Twitter Using Electric Fields Analogy*. In: *S*emantic Computing (ICSC), 2013 IEEE Seventh International Conference on. IEEE. 2013: 220–227.
[4] D'Andrea E, Ducange P, Lazzerini B, et al. Real-Time Detection of Traffic from Twitter Stream Analysis. *IEEE Trans Intell Transp Syst* 2015; 16: 2269–2283.
[5] Bifet A, Holmes G, Kirkby R, et al. Moa: Massive Online Analysis. *J Mach Learn Res* 2010; 11: 1601–1604.
[6] Morales GDF, Bifet A. SAMOA: Scalable Advanced Massive Online Analysis. *J Mach Learn Res* 2015; 16: 149–153.
[7] Wu Y. Network Big Data: A Literature Survey on Stream Data Mining. *JSW* 2014; 9: 2427–2434.
[8] Prasad BR, Agarwal S. Stream Data Mining: Platforms, Algorithms, Performance Evaluators and Research Trends. *Int J Database Theory Appl* 2016; 9: 201–218.
[9] Aggarwal CC, Reddy CK. *Data Clustering: Algorithms and Applications*. CRC press, 2013.
[10] Mousavi M, Bakar AA, Vakilian M. Data Stream Clustering Algorithms: A review. *Int J Adv Soft Comput Appl* 2015; 7: 13.
[11] Aggarwal CC, Philip SY, Han J, et al. A Framework for Clustering Evolving Data Streams. In: Proceedings 2003 VLDB Conference. Elsevier, 2003: 81–92.
[12] Aggarwal CC, Han J, Wang J, et al. A Framework for Projected Clustering of High Dimensional Data Streams. In: Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. VLDB Endowment. 2004: 852–863.
[13] Khalilian M, Mustapha N, Sulaiman N. Data Stream Clustering by Divide and Conquer Approach Based on Vector Model. *J Big Data* 2016; 3: 1.
[14] Gu X, Angelov P, Kangin D, et al. Self-Organised Direction Aware Data Partitioning Algorithm. *Inf Sci* 2018; 423: 80–95.
[15] Cao F, Estert M, Qian W, et al. Density-Based Clustering over an Evolving Data Stream With Noise. In: Proceedings of the 2006 SIAM international conference on data mining. SIAM, 2006: 328–339.
[16] Tu L, Chen Y. Stream Data Clustering Based on Grid Density and Attraction. *ACM Trans Knowl Discov Data TKDD* 2009; 3: 12.
[17] Hyde R, Angelov P, MacKenzie A. Fully Online Clustering of Evolving Data Streams Into Arbitrarily Shaped Clusters. *Inf Sci* 2017; 382: 96–114.
[18] Su J, Li Y, Zhao X. Data Stream Clustering by fast Density-Peak-Search. *Stat Interface* 2018; 11: 183–189.
[19] Rodrigues PP, Gama J, Pedroso J. Hierarchical Clustering of Time-Series Data Streams. *IEEE Trans Knowl Data Eng* 2008; 20: 615–627.
[20] Udommanetanakit K, Rakthanmanon T, Waiyamai K. E-stream: *Evolution-Based Technique for Stream Clustering*. In: International Conference on Advanced Data Mining and Applications. Springer, 2007: 605–615.
[21] Meesuksabai W, Kangkachit T, Waiyamai K. Hue-stream: *Evolution-Based Clustering Technique for Heterogeneous Data Streams With Uncertainty*. In: International Conference on Advanced Data Mining and Applications. Springer, 2011: 27–40.
[22] Zhou A, Cao F, Yan Y, et al. Distributed Data Stream Clustering: *A Fast EM-Based Approach*. In: *Data Engineering, 2007. ICDE 2007.* IEEE 23rd International Conference on. IEEE. 2007: 736–745.
[23] Dang XH, Lee VC, Ng WK, et al. *Incremental and Adaptive Clustering Stream Data Over Sliding Window*. In: International Conference on Database and Expert Systems Applications. Springer, 2009: 660–674.
[24] Barddal JP, Gomes HM, Enembreck F. SNCStream: A Social Network-Based Data Stream Clustering Algorithm. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing. ACM, 2015: 935–940.
[25] Nutakki GC, Nasraoui O. *Clustering Data Streams with Adaptive Forgetting*. In: Big Data (Big Data Congress), 2017 IEEE International Congress on, IEEE. 2017: 494–497.
[26] Domingos P, Hulten G. Mining high-speed data streams. In: *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2000, pp. 71–80.
[27] Hulten G, Spencer L, Domingos P. Mining Time-Changing Data Streams. In: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2001: 97–106.
[28] Aggarwal CC, Han J, Wang J, et al. On Demand Classification of Data Streams. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. 2004: 503–508.
[29] Mena-Torres D, Aguilar-Ruiz JS. A Similarity-Based Approach for Data Stream Classification. *Expert Syst Appl.* 2014; 41: 4224–4234.
[30] Gomes HM, Enembreck F. SAE2: Advances on the Social Adaptive Ensemble Classifier for Data Streams. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM. 2014: 798–804.

[31] Barddal JP, Gomes HM, Enembreck F. SFNClassifier: A Scale-free Social Network Method to Handle Concept Drift. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM. 2014: 786–791.
[32] Sun Y, Tang K, Minku LL, et al. Online Ensemble Learning of Data Streams with Gradually Evolved Classes. *IEEE Trans Knowl Data Eng.* 2016; 28: 1532–1545.
[33] Ren S, Liao B, Zhu W, et al. Knowledge-Maximized Ensemble Algorithm for Different Types of Concept Drift. *Inf Sci.* 2018; 430: 261–281.
[34] Ueno K, Xi X, Keogh E, et al. *Anytime Classification Using the Nearest Neighbor Algorithm With Applications to Stream Mining*. In: Data Mining, 2006. ICDM'06. Sixth International Conference on. IEEE, 2006: 623–632.
[35] Mohamad S, Sayed-Mouchaweh M, Bouchachia A. Active Learning for Classifying Data Streams With Unknown Number Of Classes. *Neural Netw* 2018; 98: 1–15.
[36] Zaharia M, Xin RS, Wendell P, et al. Apache Spark: A Unified Engine for Big Data Processing. *Commun ACM* 2016; 59: 56–65.